Université Abou Bekr Belkaid
Tlemcen Algérie
جامعة أبي بكر بلقايد
تلمسان الجزائر

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

pour l'obtention du diplôme de Licence en Informatique

Thème

Développement mobile sous Android

(Conception et réalisation d'un jeu en réseau « jeu d'allumettes »)

Réalisé par :

- ALIANE Lotfi Oussama
- AZZOUNI Ismehen.

Présenté le 10 Juin 2014 devant la commission d'examination composée de MM.

- *SMAHI M.* *(Encadreur)*
- *BENAZZOUZ M.* *(Examinateur)*
- *MESSABIHI M.* *(Examinateur)*

Année universitaire : 2013-2014

Dédicaces

A tous ceux qui nous ont soutenus tout au long de ce projet.

A nos chers parents, que nulle dédicace ne puisse exprimer ce qu'on leurs doit, pour leurs bienveillance, leur affectation et leur soutien morale durant l'élaboration de ce travail, en témoignage de notre profond amour et nos sincères reconnaissances pour les efforts qu'ils ont consenti pour l'accomplissement de nos études, on leurs dédie ce modeste travail

« Que dieu vous préserve et vous procure santé »

« اللهم اجعل عملنا هذا في ميزان حسناتهم »

AZZOUNI Ismahen
Et
ALIANE Lotfi Oussama

Remerciements

Avant d'entamer ce rapport de projet de fin d'études, nous tenons à exprimer notre sincère gratitude envers tous ceux qui nous ont aidé ou ont participé au bon déroulement de ce projet.

Tout d'abord, nous tenons à remercier notre encadreur M. SMAHI Mohammed Ismail pour sa compréhension, sa disponibilité et son aide inestimable à la mise en place de ce modeste travail.

Nous nous devons aussi remercier particulièrement trois personnes M'hammedi Islem et Khadidja, Aliane Anwar pour leurs générosité à nous procurer leur téléphone portable durant plusieurs jours d'essais ainsi que tous nos frères, sœurs et amis qui avaient la gentillesse de nous partager leurs avis.

تلخيص

خلال مشروعنا تم التحدث عن مجال التطبيقات النقالة و كان هدفنا الرئيسي الوصول الى تحقيق لعبة ذات لاعبين متواصلين بينهما عن بعد و ذلك خصيصا بالتطرق الى أحد التكنولوجيات الجديدة و هي الواي-فاي المباشر، أما الجانب النظر لهذا المشروع هو الألغوريتم Minimax الذي اعتمد عليه في تطوير لعبة الكبريت 2 AS Zalamites

Résumé

Tout au long de ce modeste travail, nous avons abordé le domaine de développement d'application mobile. Notre objectif principal, était de réaliser une application Android, sous forme d'un jeu entre deux joueurs distants. La spécificité de notre travail était dans la manière d'établir une connexion à distance, en utilisant une nouvelle technologie de communication en l'occurrence le Wi-Fi Direct. L'aspect théorique dans notre projet était l'utilisation de l'algorithme MiniMax pour développer le jeu d'allumettes baptisé : 2AS Zalamites

Mots-clés : Android, Wi-fi Direct, Algorithme MiniMax, Théorie des jeux.

Abstract

Throughout this modest work, we discussed the field of mobile application development. Our main objective was to create an Android application under form a game between two remote players. The specificity of our work was in how to establish a remote connection using a new communication technology namely the Wi-Fi Direct. the theoretical aspect in our project was the use of the MiniMax algorithm to develop the match game called: 2AS Zalamites

Keywords: Android, Wi-Fi Direct, MiniMax Algoirthm, Game Theory

Table des matières

Liste des figures

Introduction générale

Le marché de la téléphonie portable connaît actuellement une véritable révolution. Google, ayant réalisé le potentiel de ce marché, a décidé de s'y introduire en rachetant une startup travaillant sur un système d'exploitation ouvert pour terminal mobile : Android.

Dans le cadre de notre projet de développement d'application sous le Système d'exploitation Android, nous étions menées d'un coté, à explorer ce nouveau système, et de l'autre coté de réaliser et implémenter une application sous réseau ad-hoc, en utilisant la technologie Wifi-Direct. Cette application avait pour but d'implémenter le « jeu d'allumettes » qui est un jeu entre deux joueurs, en respectant les différentes notions déjà développées dans le domaine de la Théorie des jeux.

Ainsi, nous articulons notre rapport autour de quatre chapitres :

Le premier chapitre portera sur la Programmation mobile sous Android. Le deuxième chapitre aura comme but, Analyse et conception de l'application « 2AS Zalamites ». La troisième partie portera sur le Mode de communication Wifi-Direct. Finalement, nous décrirons dans le dernier chapitre le fonctionnement de notre application.

Chapitre 1 : Programmation mobile sous Android

I. Introduction

Ces dernières années l'utilisation des applications « intelligentes » sur les Smartphones et tablettes deviennent de plus en plus fréquentes et cela grâce aux différents systèmes d'exploitation mobile tel que : « Android », « IOS[1] » ou bien «Windows Phone » qui ne cessent de se développer.

Ce chapitre sera scindé en deux parties distinctes : dans la première partie nous nous intéressons au développement mobile sous Android, en détaillant son historique, ces caractéristiques ainsi que son architecture. Par contre, la deuxième partie sera dédiée pour faire une petite introduction à la théorie des jeux, en développant plus spécifiquement l'algorithme MinMax

II. Alors qu'est-ce qu'Android

1. Introduction

Android est un système d'exploitation pour tablettes et smartphone, aussi appelé téléphone intelligent ou Ordiphone. Ce projet est né d'un consortium de 34 entreprises (en 2008), initié par la société Google le 5 novembre 2007, et appelé l'Open Handset Alliance ou OHA. Son but consistait à trouver une solution fiable pour concurrencer Apple avec l'IphoneOS, Microsoft avec Windows Mobile, Nokia avec Symbian et Research In Motion avec Blackberry OS. En effet à cette époque, la quasi-totalité des smartphones fonctionnait sur ces systèmes d'exploitation [**1**].

Cependant, la principale différence d'Android face aux autres solutions est qu'il soit open source. Cette particularité le rend donc gratuit, et personnalisable par les constructeurs et les opérateurs de téléphonie mobile. Malgré des personnalisations accrues en terme d'interface utilisateur et de fonctionnalité, le cœur du système reste commun ce qui permet une interopérabilité des applications [**1**]. De ce fait, si l'on ne prend pas en compte les difficultés liées aux différences matérielles des périphériques, deux téléphones issus de différents constructeurs qui fonctionnent tous les deux sur Android, seront en mesure d'exécuter les mêmes applications sans aucune compilation supplémentaire.

[1] IOS : Iphone Operating System

2. Historique d'Android

En juillet 2005, Google a acquis Android, Inc., une petite startup qui développait des applications pour téléphones mobiles .C'est à ce moment-là que des rumeurs sur l'entrée de Google dans le secteur du mobile ont commencé. Mais personne n'avait des données sûres à propos des marchés dans lesquels ils allaient se positionner.

Après ce rachat fait par Google, une équipe dirigée par Andy Rubin, un ancien d'Android Inc, a commencé à travailler sur un système d'exploitation pour appareil mobile basé sur linux. Durant 2 ans, avant que l'OHA[2] soit créer officiellement, un certain nombre de rumeurs ont circulé au sujet de Google. Il a été dit que Google développait des applications mobiles de son moteur de recherche, qu'elle développait un nouveau téléphone mobile, etc.

En 2007, le 5 novembre, l'OHA a été officiellement annoncée, ainsi que son but : développer des standards open sources pour appareil mobile [2]. Le premier standard annoncé a été Android, une plateforme pour appareils mobiles basée sur un kernel linux 2.6.

Figure 1: Evolution des versions Android[5]

[2] l'Open Handset Alliance

En octobre 2008, apparait la première version d'Android qui n'avait pas reçu de nom. Cette version s'est avérée être la β du système.

La version 1.5 Cupcake corrigea le manque d'API et rendit le système plus utilisable. Depuis, Android 1.6, 2.0 et 2.1 ont apporté d'importantes améliorations respectivement sur les fonctionnalités et sur l'interface graphique du système (Voir Figure 1).

Android 2.2 Froyo a fortement mis l'accent sur la synergie avec Internet. L'envoi d'applications et de liens instantanés depuis un ordinateur est désormais possible. Aussi, Google annonce-t-elle que le navigateur chrome intégré à Android 2.2 est le navigateur mobile le plus rapide au monde grâce à l'intégration du moteur JavaScript V8.

Android 3.0 Honeycomb est spécialement étudié pour les tablettes tactiles. Les premiers modèles devraient être annoncés au CES 2011.

On y apprend quelques nouveautés comme la prise en charge de la vidéo-conférence via Gtalk, la nouvelle interface Gmail ou encore le lecteur de livre électronique Google.

La refonte graphique de l'interface utilisateur est assez réussie, plus d'informations devraient suivre dont surement des éclaircissements sur l'intégration ou non de l'interface de cette version d'Android sur les futurs smartphones.

Android 4.0 devrait arriver très vite (mi 2011) pour rajouter encore plus de fonctionnalités aux terminaux. Pour le développement, ces nouvelles versions d'Android devraient proposer de nouveaux composants permettant de réaliser des applications avec une ergonomie plus adaptée aux tablettes tactiles.

Android 3.0 et Android 4.0 devraient apporter plus d'outils aux constructeurs leur permettant de proposer des tablettes tactiles, qui seront capables de rivaliser (surtout au niveau de l'ergonomie) avec Ipad.

III. Présentation du SDK3

1. La plateforme SDK

Le SDK Android est un outil permettant de programmer en Java pour les terminaux Android. Ce SDK utilise donc les mécanismes Fondamentaux de Java (Héritage simple, interface,

3 Software Development Kit

classe, ...), tout en ayant des particularités vis à vis de la construction d'une Application. Nous devons par exemple bien séparer les classes graphiques des classes du modèle. Les classes correspondant à des changements de vues sont appelées Activity. L'emplacement des boutons et des images dans une vue est déclaré dans un fichier XML appelé Layout. Ce fichier sera appelé par la classe Activity.

Le cycle de vie d'une classe « Activity » est représenté la figure qui suit :

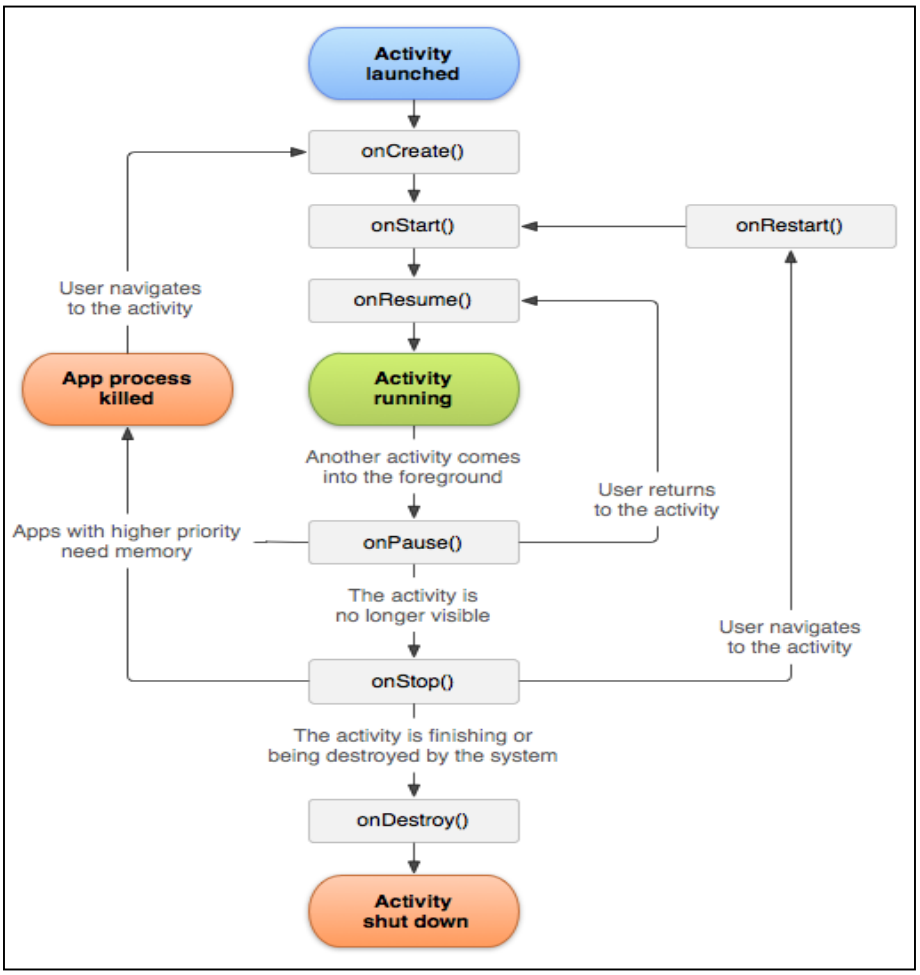


Figure 2 : Cycle de vie d'une classe Activity

Nous remarquons d'après ce diagramme, qu'on fait appel à la méthode `onCreate()` lors d'un changement de vue. Lorsque l'on revient à une vue précédente, on fait appel à la méthode `onDestroy()` de l'Activity en cours. Ainsi les données présentes dans une Activity ne seront pas persistantes au cours de l'utilisation de l'application.

Le système Android a la spécificité suivante : chaque application doit être munie d'un fichier `AndroidManifest.xml`. Ce fichier XML4 contient toutes les autorisations nécessaires pour permettre à l'application d'utiliser la fonctionnalité d'Android, comme l'accès au Wi-Fi ou au Bluetooth. Ce fichier doit Contenir toutes les Activity de l'application dont l'Activity principale qui sera appelée au démarrage de l'application

Un aperçu du fichier AndroidManifest.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.android.wifidirect.discovery"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="16" />

    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
    <uses-permission android:name="android.permission.CHANGE_NETWORK_STATE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.VIBRATE" />

    <!-- Google Play filtering -->
    <uses-feature
        android:name="android.hardware.wifi.direct"
        android:required="true" />
```

Figure 3: Aperçu du fichier XML

Une autre particularité du système Android est la facilité de connexion en Bluetooth et en Wi-Fi à un autre système.

Pour la suite de notre travail, nous avons opté pour le Wi-Fi direct comme protocole de communication.

2. Avantages du SDK

Les possibilités offertes par le SDK regroupent, entre autres, l'accès à toutes les fonctionnalités de l'appareil comme l'accès à Internet, à la partie téléphonie (appel, gestion des contacts et des SMS), à l'appareil photo, au GPS, à l'accéléromètre ou encore à la boussole numériques.

4 eXtensible Markup Langage

Les opportunités de marché du développement sur Android, sont extrêmement nombreuses et de nouveaux concepts sont découverts au fur et à mesure que les appareils sont enrichis en divers capteurs et que le SDK est mis à jour.[1]

Les applications mobiles sont à leurs débuts et ont toutes les chances de devenir incontournables dans un futur proche. Qui pourrait trouver inutile les fameux annuaires téléphoniques par exemple ? N'est-ce pourtant pas beaucoup plus pratique d'avoir le contenu de ses centaines de numéros dans votre poche ? Ajoutons à la simple fonctionnalité de recherche une géo-localisation et la possibilité aux utilisateurs de laisser des commentaires et la prochaine fois que vous serez à la recherche d'un restaurant en vacances vous ne vous contenterez pas d'un fastfood... Beaucoup d'experts s'accordent sur le fait que l'informatique mobile est l'avenir de l'informatique grand publique. Certes les supports seront multiples, il sera difficile de se contenter d'un écran de quelques pouces pour une utilisation intensive.

Cependant, la plupart des utilisateurs d'ordinateur et d'internet font des tâches ne nécessitant pas un poste de travail comme nous le connaissons. Une tablette tactile d'une dizaine de pouces de diagonale conviendrait très bien à leurs utilisations si les applications existaient.

IV. L'importance des applications dans système Android

Comme tout système d'exploitation l'absence des applications rend l'appareil sans intérêt, donc le système Android fonctionne avec plusieurs applications conçues par Google ou d'autre développeurs et permettant une utilisation complète et facile d'un Smartphone. Ce type d'architecture donne aussi la possibilité aux développeurs et le choix à l'utilisateur final d'intégrer les applications désirées selon le besoin dans leurs Smartphone.

1. Architecture Android

La figure suivante (Figure 4) suivant illustre les composants principaux du système d'exploitation Android. Chaque section sera décrite dans ce qui suit :

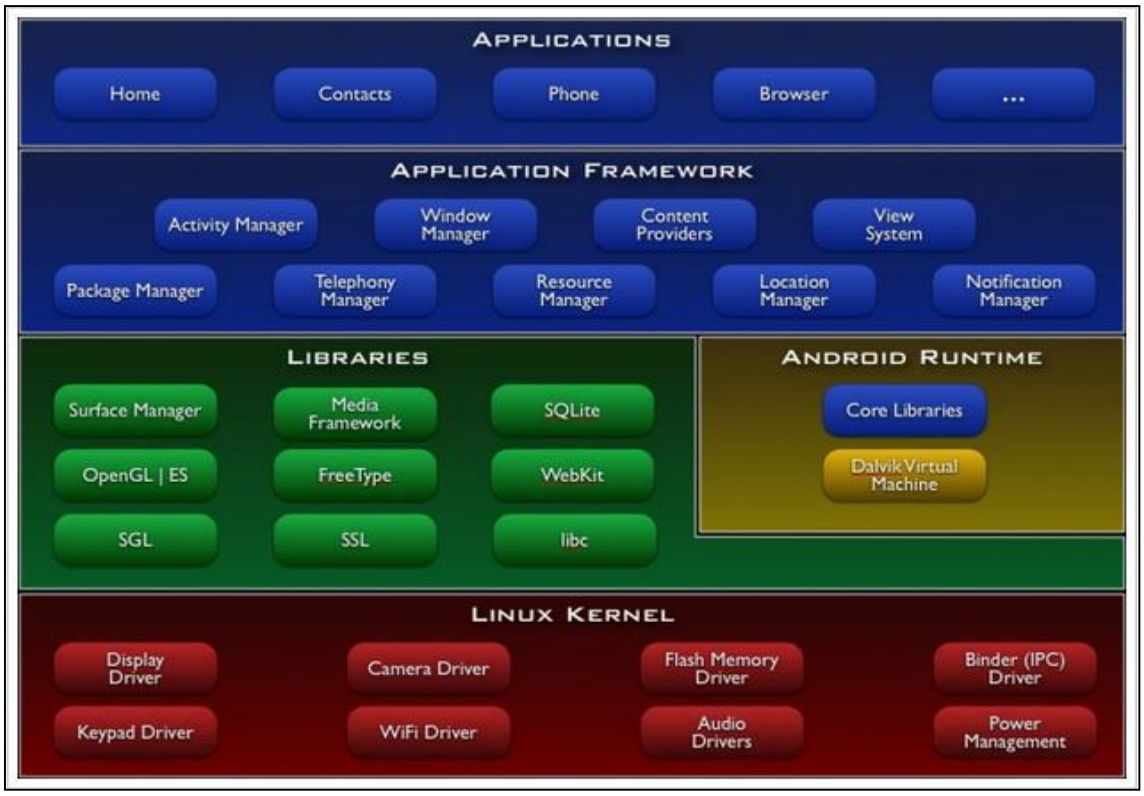


Figure 4 : Les composants du système Android [2]

2. Applications

Android est fourni avec un ensemble d'applications dont un client email, une application SMS, un calendrier, un service de cartographie, un navigateur… toutes écrites en JAVA.

3. Framework de développement

En fournissant une plateforme de développement ouverte, Android offre aux développeurs la possibilité de créer des applications extrêmement riches et innovantes. Les développeurs sont libres de profiter du matériel périphérique et informations sur la localisation d'accès, exécuter des services d'arrière-plan, définir des alarmes, ajouter des notifications à la barre d'état, etc.

Les développeurs ont un accès complet au même Framework API utilisé par les applications de base. L'architecture d'application est conçue pour simplifier la réutilisation des composants; n'importe quelle application peut publier ses capacités et n'importe quelle autre application peut alors faire usage de ces capacités (soumis à des contraintes de sécurité appliquées par le Framework). Ce même mécanisme permet aux composants d'être remplacés par l'utilisateur.

Toutes les applications sous-jacentes forment un ensemble de services et de systèmes, y compris:

- Un jeu extensible de vues qui peuvent être utilisées pour construire une application.
- Des fournisseurs de contenu qui permettent aux applications d'accéder aux données d'autres applications (telles que les Contacts), ou de partager leurs propres données
- Un gestionnaire de ressources.
- Un gestionnaire de notification qui permet à toutes les demandes d'afficher des alertes personnalisées dans la barre d'état.
- Un gestionnaire d'activité qui gère le cycle de vie des applications et propose une navigation commune.

4. Bibliothèques

Android dispose d'un ensemble de librairies C / C++ utilisées par les différents composants du système Android. Elles sont offertes aux développeurs à travers le Framework Android. En voici quelques-unes:

Système de bibliothèque C – une mise en œuvre dérivée de BSD de la bibliothèque C standard du système (libc), destinés aux systèmes embarqués basés sur Linux.

Comme cela a été dit précédemment, Android ne supporte pas la glibc, donc les ingénieurs d'Android ont développé une librairie C (libc) nommé Bionic libc . Elle est optimisée pour les appareils mobiles et a été développée spécialement pour Android.

Les ingénieurs d'Android ont décidé de développer une libc propre à la plateforme Android car ils avaient besoin d'une libc légère (la libc sera chargée dans chaque processus) et rapide (les appareils mobiles ne disposent pas de CPU puissant).

La Bionic libc a été écrit pour supporter les CPU ARM, bien que le support x86 soit présent. Il n'y pas de support pour les autres architectures CPU telles que PowerPC ou MIPS. Néanmoins, pour le marché des appareils mobiles, seulement l'architecture ARM est importante. Cette libc est sous licence BSD. Elle reprend une grande partie du code des glibc issue d'OpenBSD, FreeBSD et NetBSD.

Ces caractéristiques importantes :

- Elle pèse environ 200Ko, soit la moitié de la glibc
- L'implémentation des pthreads (POSIX thread) a été complètement réécrite pour supporter les threads de la machine virtuelle Dalvik. De ce fait, la Bionic libc ne supporte pas les threads POSIX
- Les exceptions C++ et les "wide char" ne sont pas supportés
- Médiathèques – basée sur PacketVideo de OpenCore; les librairies permettant la lecture et l'enregistrement audio et vidéo, ainsi que la gestion des fichiers image, y compris MPEG4, H.264, MP3, AAC, AMR, JPG et PNG.

Le schéma ci-dessous décrit tous les éléments de l'architecture de ces médiathèques:

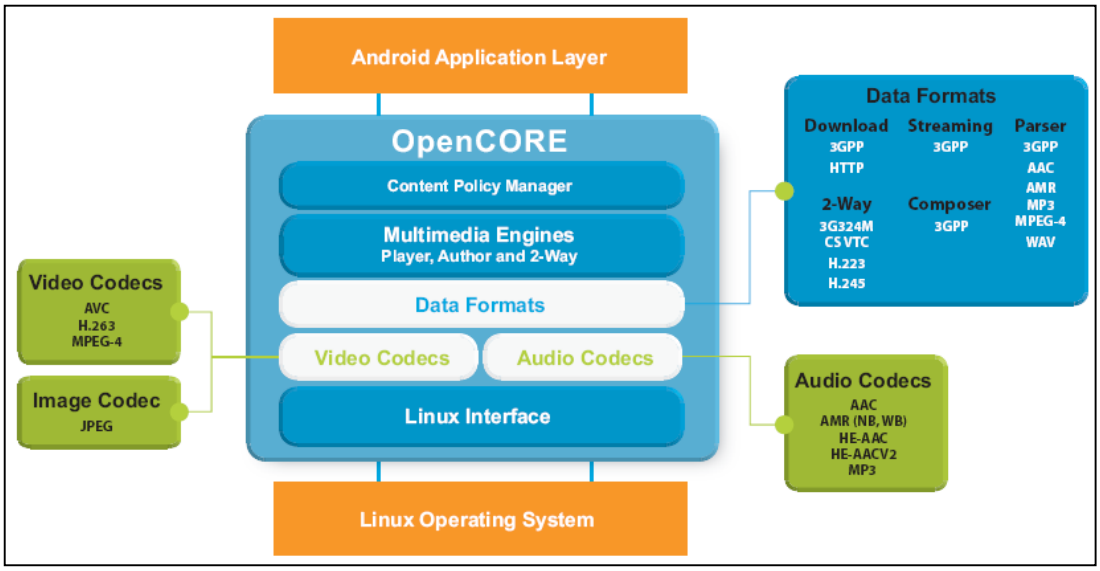


Figure 5: architecture de ces médiathèques

- Surface Manager – gère l'accès au sous-système d'affichage et de façon transparente.
- LibWebCore – Le navigateur web présent dans Android est basé sur le moteur de rendu sous licence BSD WebKit.
- WebKit est moteur de rendu, qui fournit une "fondation" sur laquelle on peut développer un navigateur web. Il a été originellement dérivé par Apple du moteur de rendu KHTML pour être utilisé par le navigateur web Safari et maintenant il est développé par KDE project, Apple, Nokia, Google et d'autres. WebKit est composé de deux librairies : WebCore et JavascriptCore qui sont disponibles sous licence GPL.

WebKit supporte le CSS, JavaScripts, DOM, AJAX. La dernière version a obtenu 100% au test Acid 3. La version de WebKit présent dans Android à été légèrement modifiée pour s'adapter aux appareils mobiles. Ainsi, le moteur de rendu basé sur WebKit présent dans Android supporte l'affichage sur une colonne.

- SGL – le moteur graphique 2D.
- Bibliothèques 3D – une implémentation basée sur OpenGL ES 1.0 API; les bibliothèques utilisent l'accélération 3D matérielle (si disponible).
- FreeType – bitmap et vectoriel de rendu de police.
- SQLite – un moteur de base de données relationnelles puissant et léger, disponible pour toutes les applications.

5. Android Runtime

Android inclut un ensemble de librairies de base offrant la plupart des fonctionnalités disponibles dans les bibliothèques de base du langage de programmation Java.

Chaque application Android s'exécute dans son propre processus, avec sa propre instance de la machine virtuelle Dalvik. Dalvik a été écrit pour que le dispositif puisse faire tourner plusieurs machines virtuelles de manière efficace. La machine virtuelle Dalvik exécute des fichiers dans l'exécutable Dalvik (. DEX), un format optimisé pour ne pas encombrer la mémoire. La machine virtuelle est la base de registres et fonctionne grâce aux classes compilées par un compilateur Java et transformées dans le format DEX.

La machine virtuelle Dalvik s'appuie sur le noyau Linux pour les fonctionnalités de base telles que le filetage et la gestion de la mémoire de bas niveau.

6. Linux Kernel

Android est basé sur un kernel linux 2.6 mais ce n'est pas linux. Il ne possède pas de système de fenêtrage natif (X windows system). La glibc n'étant pas supportée, Android utilise une libc customisée appelée Bionic libc.

Enfin, Android utilise un kernel avec différents patches pour la gestion de l'alimentation, le partage mémoire, etc. permettant une meilleure gestion de ces caractéristiques pour les appareils mobiles.

Android n'est pas linux mais il est basé sur un kernel linux. Pourquoi sur un kernel linux?

Le kernel linux a un système de gestion mémoire et de processus reconnu pour sa stabilité et ses performances.

Le model de sécurité utilisé par linux, basé sur un système de permission, est connu pour être robuste et performant. Il n'a pas changé depuis les années 70

- Le kernel linux fournit un système de driver permettant une abstraction avec le matériel. Il permet également le partage de librairies entre différents processus, le chargement et le déchargement de modules à chaud.
- le kernel linux est entièrement open source et il y a une communauté de développeurs qui l'améliorèrent et rajoutent des drivers.

C'est pour les points cités ci-dessus que l'équipe en charge du noyau a décidé d'utiliser un kernel linux.

- le kernel linux est entièrement open source et il y a une communauté de développeurs qui l'améliorèrent et rajoutent des drivers.

C'est pour les points cités ci-dessus que l'équipe en charge du noyau a décidé d'utiliser un kernel linux.

V. Théorie des jeux

1. Introduction

Une activité, certes ludique, pour laquelle les êtres humains déploient leur intelligence depuis la nuit des temps est le jeu. Il est donc naturel de chercher de créer des programmes de jeux en utilisant des méthodes de l'intelligence artificielle. On pourrait même évaluer les progrès de l'intelligence artificielle avec les progrès des programmes de jeux.

Les jeux offrent en effet un paradigme exceptionnel pour les méthodes de l'intelligence artificielle. Ils ont un nombre de règles limité et des situations parfois très complexes. De plus les décisions, lors du déroulement d'un jeu, peuvent être évaluées de façon exacte, ce qui permet d'avoir une idée précise sur la valeur du programme de jeu.

2. L'algorithme MiniMax

Dans le cadre de notre projet, nous nous sommes intéressé aux jeux à deux joueurs avec informa²tion parfaite, i.e. chaque joueur est au courant de ce que l'autre joueur a fait ou peut en faire. Le joueur qui commence s'appelle le joueur MAX (on dit aussi le joueur A) et l'autre joueur s'appelle MIN (ou le joueur B). Ceci parce que :

- celui qui commence cherche à trouver, parmi toutes les situations qui a sa disposition, une situation qui lui permet de maximiser ses gains ;
- celui qui répond doit trouver, à partir de toutes les situations qui conduisent à la victoire du premier joueur, la situation qui minimise les gains de ce joueur.

3. Principe

Le principe consiste à construire un arbre de jeu : c'est un arbre qui représente toutes les évolutions possibles du jeu à partir de la configuration actuelle. Comme le jeu est à information complète, toutes les informations concernant les différentes situations du jeu, les gains et les pertes sont parfaitement connues à chaque moment.

Nous allons détailler le principe de cet algorithme par le déploiement de l'arbre de jeu d'un jeu standard à savoir le Jeu de Nim ou jeu d'allumettes.

Voici un exemple de l'arbre MINIMAX pour la situation avec 5 allumettes sur la table et le tour du joueur MAX est présenté dans la figure suivante (Figure 5), où dans les triangles sont inscrits les nombres des allumettes se trouvant sur la table avant le coup.

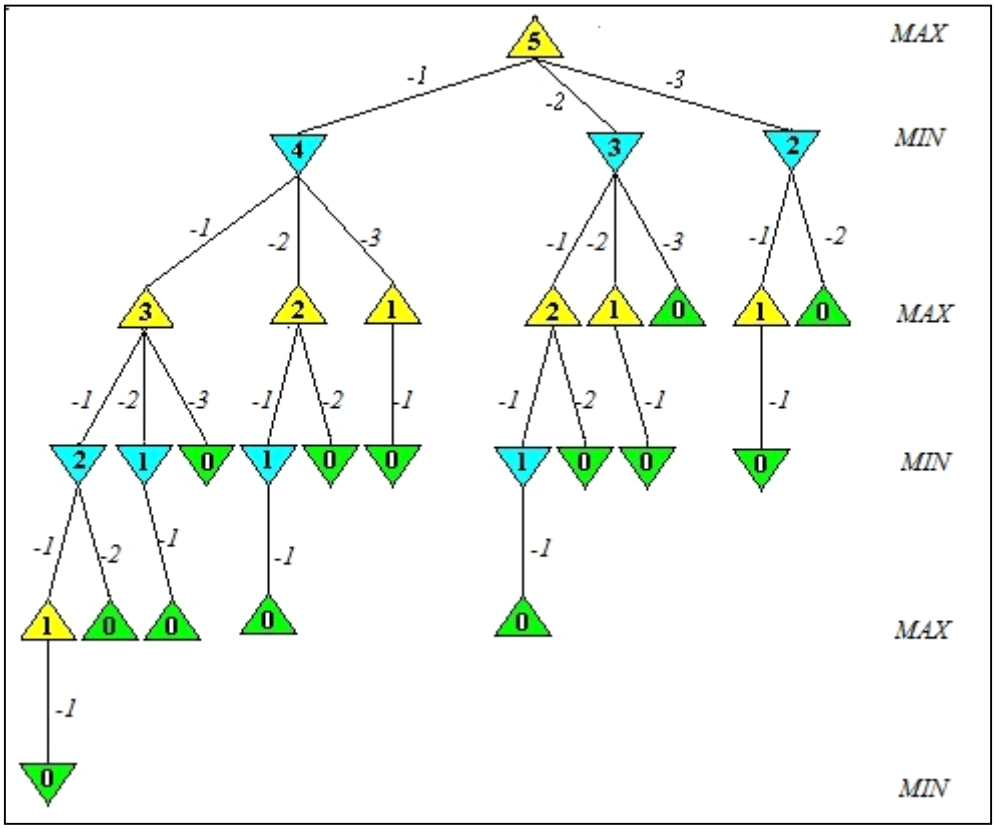


Figure 6: L`arbre MiniMax pour le jeu des allumettes [3]

Les opérateurs qui modélisent les coups sont -1, -2, -3 (le nombre des allumettes qui sont pris par l'agent) et ils sont toujours appliqués dans cet ordre, de gauche à droite dans l'arbre de jeu présenté. La fonction d'évaluation statique associe à une position terminale victorieuse pour l'agent MAX la valeur +100, et -100 pour une position perdante.

Si on détermine la valeur MiniMax pour la situation avec 5 allumettes sur la table et le tour du joueur MAX, on va trouver +100, donc le joueur MAX va gagner la partie meme si le joueur MIN joue de façon logique. On peut observer que les valeurs associées à tous les nœuds MIN successeurs de la racine sont égales à +100, donc quel que soit le coup choisit par le joueur MIN, on va arriver dans une position gagnante pour le joueur MAX.

En effet, si l'agent MAX retire 1 allumette, l'agent MIN en retire à son tour 3, si MAX en retire 2, MIN en retire 2 aussi et si MAX en retire 3, MIN va en retirer 1, et dans toutes ces situations MIN va trouver sur la table 1 allumette qu'il sera forcé de prendre, donc il gagnera la partie.

4. L'algorithme

L'algorithme de Min-Max peut être décrit comme suit :

```
function minimax(node, depth, maximizingPlayer)
    if depth = 0 or node is a terminal node
        return the heuristic value of node
    if maximizingPlayer
        bestValue := -∞
        for each child of node
            val := minimax(child, depth - 1, FALS
            bestValue := max(bestValue, val);
        return bestValue
    else
        bestValue := +∞
        for each child of node
            val := minimax(child, depth - 1, TRUE
            bestValue := min(bestValue, val);
        return bestValue

(* Initial call for maximizing player *)
minimax(origin, depth, TRUE)
```

VI. Conclusion

Dans ce chapitre, nous avons fait une étude du système d'exploitation Android en présentant ces fonctionnalités son architecture ainsi que les principaux composants du système. Nous avons aussi détaillé l'algorithme MiniMax que nous allons utiliser par la suite de notre travail.

La partie qui suit est consacrée la partie qui suit à l'analyse et à la conception de notre application, qui consiste à implémenter un jeu entre deux joueurs distants en utilisant en utilisant le Wi-fi Direct comme moyen de communication.

Chapitre 2 : Analyse et Conception du 2AS Zalamites

I. Introduction

Apres avoir vu en quoi consiste le cadre de notre application, nous introduisons dans le présent chapitre, la notion même de l'analyse et de la conception orientée objets de notre système Le but est de prévoir, d'étudier, de collecter les informations du système afin de gérer son cycle de vie d'où l'approche objet est devenu une solution technologique incontournable, c'est un réflexe quasi-automatique dès lors qu'on cherche à concevoir des logiciels complexes qui doivent résister à des évolutions incessantes. [7].

Pour cela nous utilisons le langage UML qui permet de représenter des concepts abstraits (graphique), de limiter les ambigüités (vocabulaire précis, indépendant des langages orientés objets) et de modéliser les applications (selon une vision objets).

La modélisation UML que nous avons adopté, montre les différents acteurs du système ainsi que les rôles qu'ils peuvent tenir.

II. Qu'est-ce qu'un modèle ?

Un modèle est avant tout une représentation abstraite du monde réel ce qui signifie cherché à comprendre ce qui se passe, ne pas se contenter d'une solution empirique.

L'approche orienté objet est une façon d'aborder un problème et de le découper en petits sous-problèmes. On commence par rechercher les objets du système logiciel et un modèle du système.

Un modèle va donc nous servir à communiquer et échanger des points de vue afin d'avoir une compréhension commune et précise d'un même problème [7].

III. Pourquoi modéliser ?

I. Les niveaux d'abstraction

Pour traiter un problème complexe nous devons le découpé en sous problèmes et analyser chacun de ces sous problèmes à part, jusqu'à arriver à comprendre tous les détails, ceci est le raisonnement même des niveaux d'abstraction.[9]

II. Un outil pour documenter

Si un model sert à comprendre le monde réel comme mentionner ci-dessus, il est aussi un moyens de poser ses idées pour les développeurs de l'équipe projets ou l'équipe de maintenance.

Cette documentation va assurer l'homogénéité et facilite le passage d'un projet a un autre.

III. Générer le code

En plus d'une documentation pour la maitrise de l'ouvrage, les modèles vont être utiles pour l'écriture du code, et vont nous permettre de nous concentrer sur les algorithmes et sur une analyse « haut niveau » du problème.

IV. Modélisation objet

Afin d'optimiser la compréhension ainsi que la réalisation du projet il est préférable de recourir à la méthodologie de développement RUP (Rational Unified Process), conçut pour les logiciels orientés objets, RUP a la visibilité sur tout le système d'information il leur apporte une réponse aux contraintes de changement continuel imposées. En ce sens, il renforce le contrôle sur les capacités d'évolution et de correction de tels systèmes. [6]

Cependant il nous paraît difficile d'envisager le processus UP sans recourir à UML comme support.

I. Le langage UML[5]

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue.

II. Caractéristiques de l'UML

- UML est un support de communication qui facilite la compréhension et la représentation des solutions, grâce à l'aspect formel de sa notation graphique.
- Langage universel grâce à son indépendance des langages de programmation, domaines d'application, aux processus.

[5] Unified Modeling Language

- Il repose sur un méta-model c'est-à-dire qu'il normalise la sémantique des concepts qu'il véhicule.
- Il permet de parler un langage commun normalisé mais accessible car visuel.
- Il cadre l'analyse permettant de concevoir une solution objet de manière itérative, grâce aux diagrammes qui supportent l'abstraction.

V. Conception détaillée de notre Application

Dans cette partie nous allons enfin pouvoir répondre à la question : quelles sont les fonctionnalités de notre système ? Pour cela on va utiliser plusieurs diagrammes d'UML.

1. Présentation des acteurs

En effet dans notre cas, et malgré que nous allons développer une application à deux joueurs, à savoir le joueur Min et le joueur Max, nous avons pu énumérer qu'un seul acteur « Joueur » qui généralise les deux acteurs impliqués dans notre système.

Le joueur

- Envoyer des messages (chat)
- Envoyer l'invitation de jeu
- Demander la connexion par Wifi directe
- Proposer un nombre d'allumettes
- Déroulement de la partie (retirer 1, 2 ou 3 allumettes)
- Contrôler la musique du jeu

2. Les diagrammes des cas d'utilisation

L'objectif fondamental de cette étape est d'identifier les principaux cas d'utilisation.
Nous nous intéressons donc, dans cette partie, à la réalisation des diagrammes des cas d'utilisations. Ces diagrammes sont un ensemble des actions réalisées par le système en réponse à une action d'un acteur [7].

3. Description des cas d'utilisation

L'étude des cas d'utilisation permet de structurer les besoins de l'utilisateur et les objectifs correspondant d'un système.

- **Description du cas d'utilisation Principal : « 2AS Zalamites »**

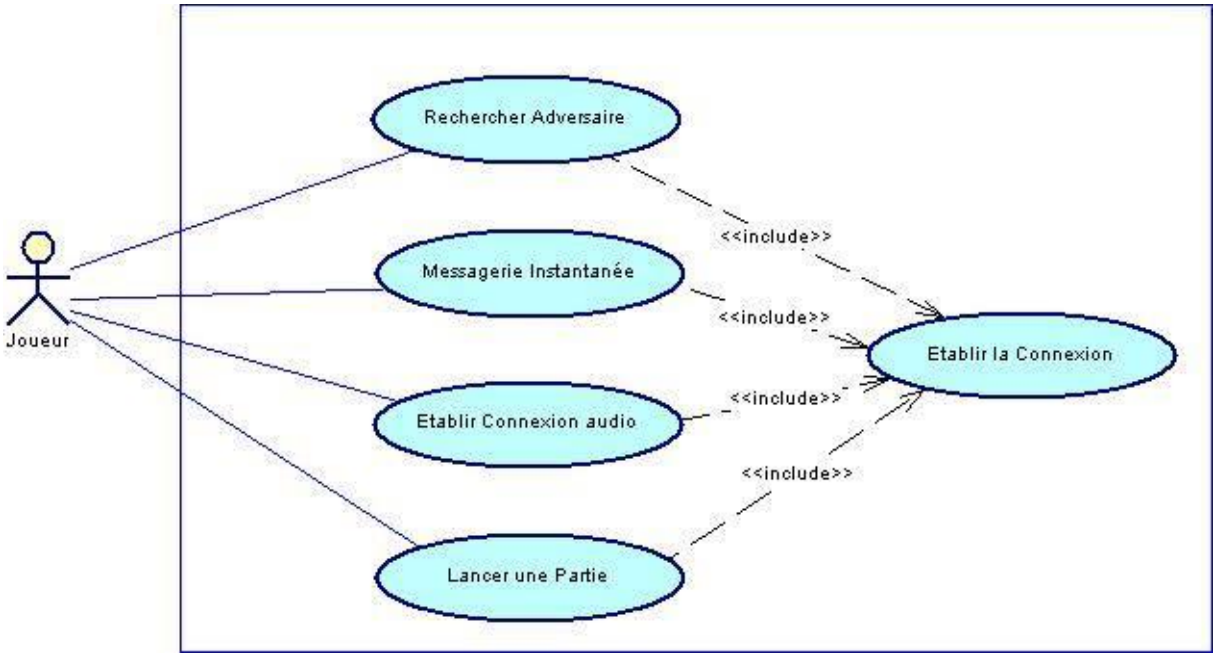


Figure 7 : Cas d'utilisation Principal

Nom du cas d'utilisation	**2AS Zalamites**
Objectif	**Utilisation générale du système**
Acteur principal	**Acteur secondaire**
Joueur Max	Joueur Min
Pré-condition	**Post-condition**
Disponibilité d'au moins 1 SmartPhone	
Déroulement Normal	
• Le joueur recherche un adversaire • Si joueur trouvé et connexion établie, nous pourrons établir une session de chat • Si joueur trouvé et connexion établie, nous pourrons établir une session de communication audio • Si joueur trouvé et connexion établie, nous pourrons initialiser et déclencher une partie	

❖ **Description du cas d'utilisation : « Lancer une partie »**

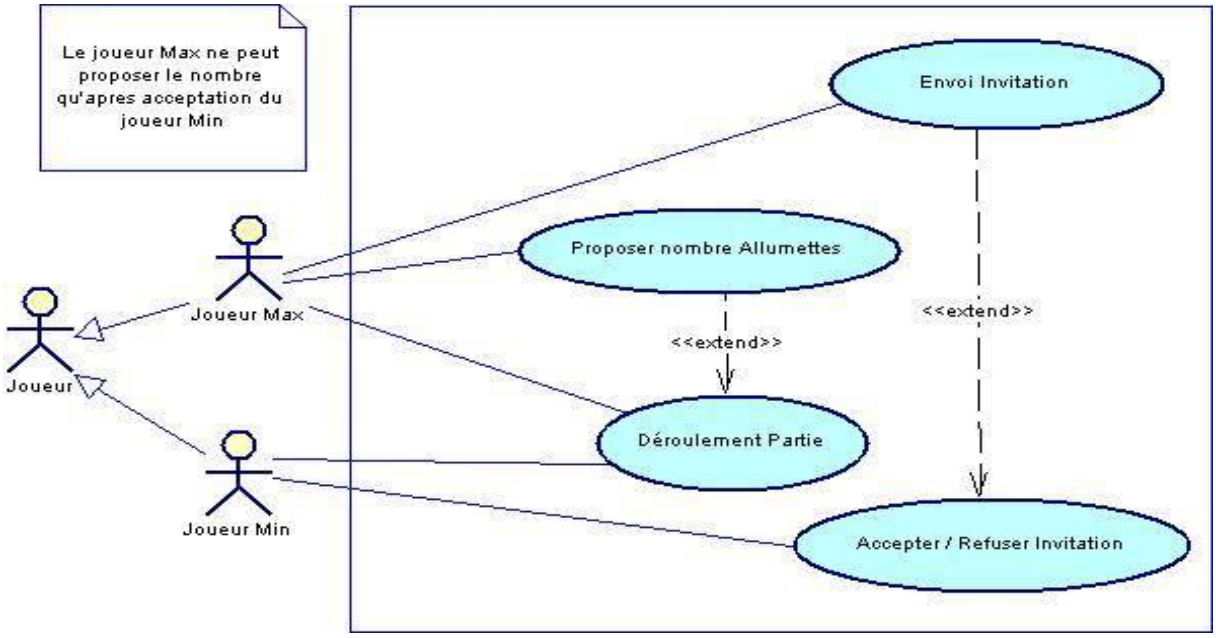


Figure 8 : Cas d'utilisation Lancer une partie

Nom du cas d'utilisation	**Lancer une partie**
Objectif	**Initialiser et déclencher le jeu**
Acteur principal	**Acteur secondaire**
Joueur Max	Joueur Min
Pré-condition	**Post-condition**
Les deux Smartphones sont connectés entre eux par Wifi directe	Lancement du jeu
Déroulement Normal	
• Le joueur Max envois une invitation au joueur Min • Le joueur Min a la possibilité soit d'accepter ou de refuser l'invitation • Si joueur Min accepte, le joueur Max doit proposer un nombre total d'allumettes • Une fois partie déclencher les deux joueurs tire alternativement 1,2 ou 3 allumettes jusqu'à épuisement du nombre total.	

4. Les diagrammes de séquences

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML.

❖ Diagramme de Connexion

Le diagramme de séquence ci-dessous, présente de point de vue chronologique, l'action d'établir une connexion entre deux joueurs (via leur SmartPhones). En effet, le joueur doit déclencher la recherche de d'autres joueurs connectés, cette action de découverte se déroule en boucle, cette dernière s'arrête si le délai de recherche est atteint sinon si un joueur distant vient d'être détecté. Si nous considérant le dernier cas, les deux joueurs peuvent commencer une session de chat.

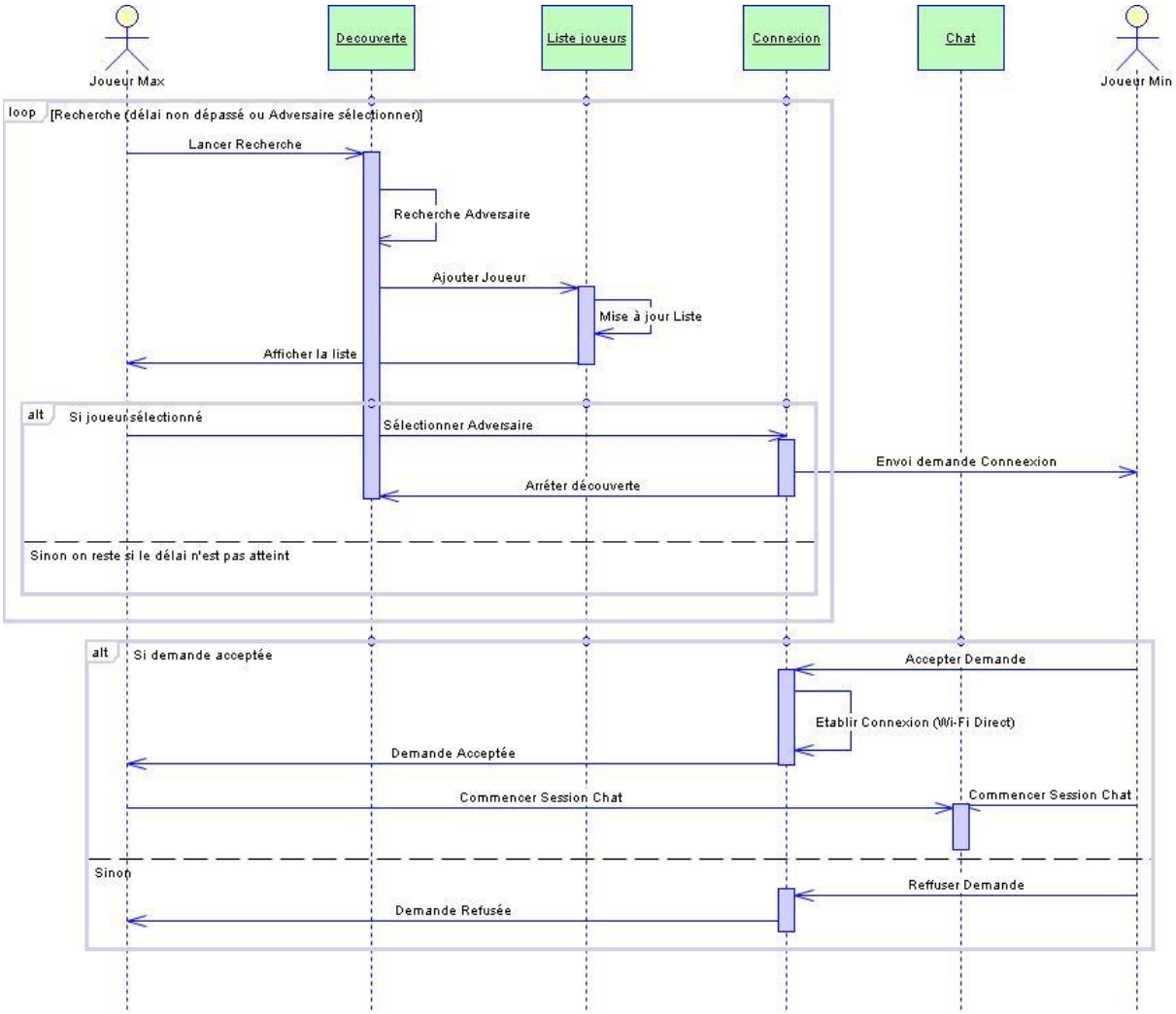


Figure 9 : Diagramme de séquence "Etablir Connexion"

❖ Diagramme de déroulement

Le diagramme de séquence « déroulement de la partie », et comme son nom l'indique, il permet de représenter tous les détails sur l'interaction qu'on peut avoir lors du déroulement d'une partie de jeu.

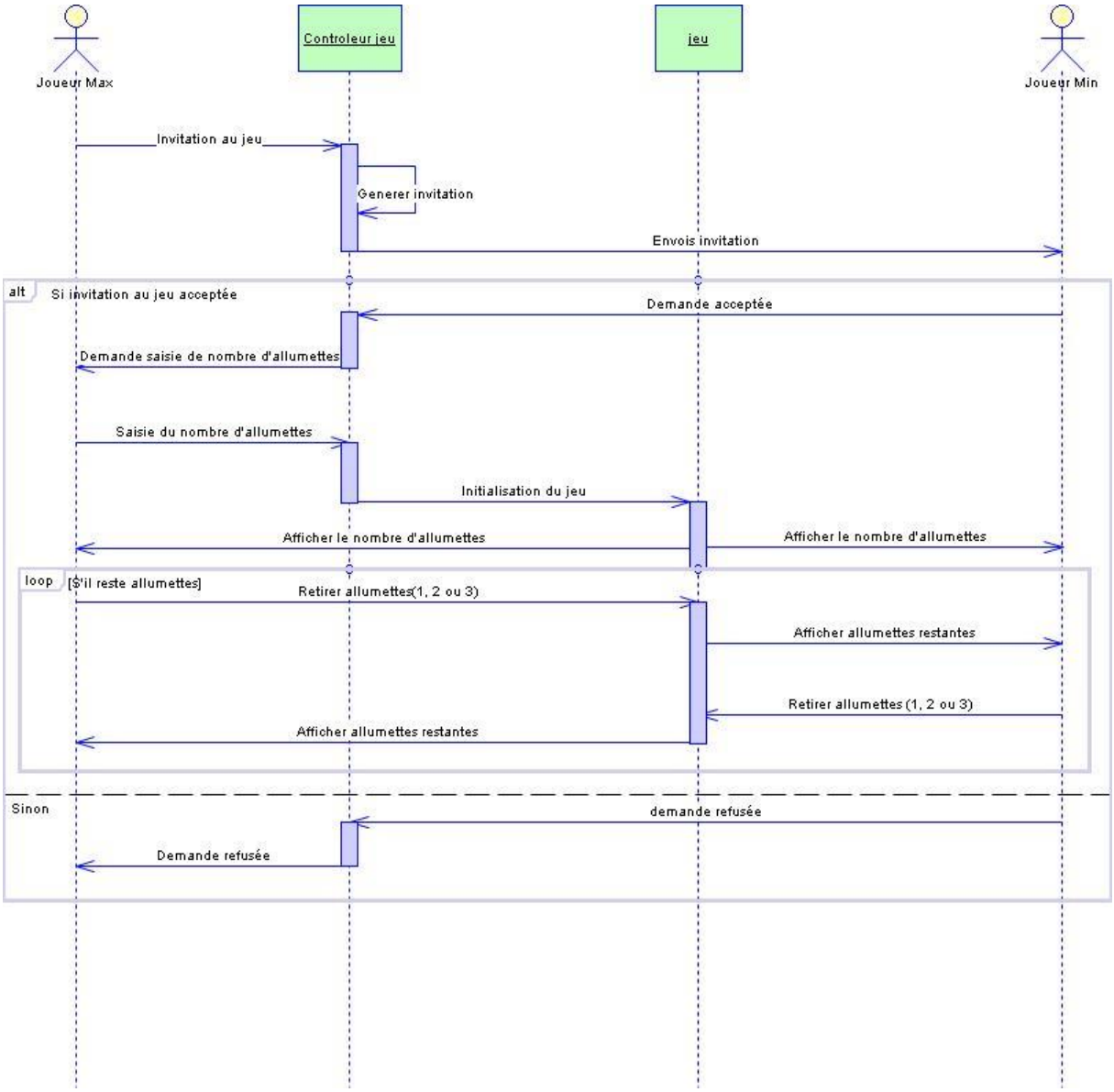


Figure 10 : Diagarmme de séquence "Déroulement d'une partie"

5. Diagramme de classes

Une classe représente la structure d'un objet, c'est-à-dire la déclaration de l'ensemble des entités qui le composent. Elle est constituée d'attributs dont les valeurs représentent l'état de l'objet et des méthodes qui sont les opérations applicables aux objets.

A partir de ce qui a été présenté tout au long de ce chapitre nous avons pu établir le diagramme de classes, ce diagramme a été raffiné après plusieurs itérations.

En effet, notre application comporte les classes suivantes :

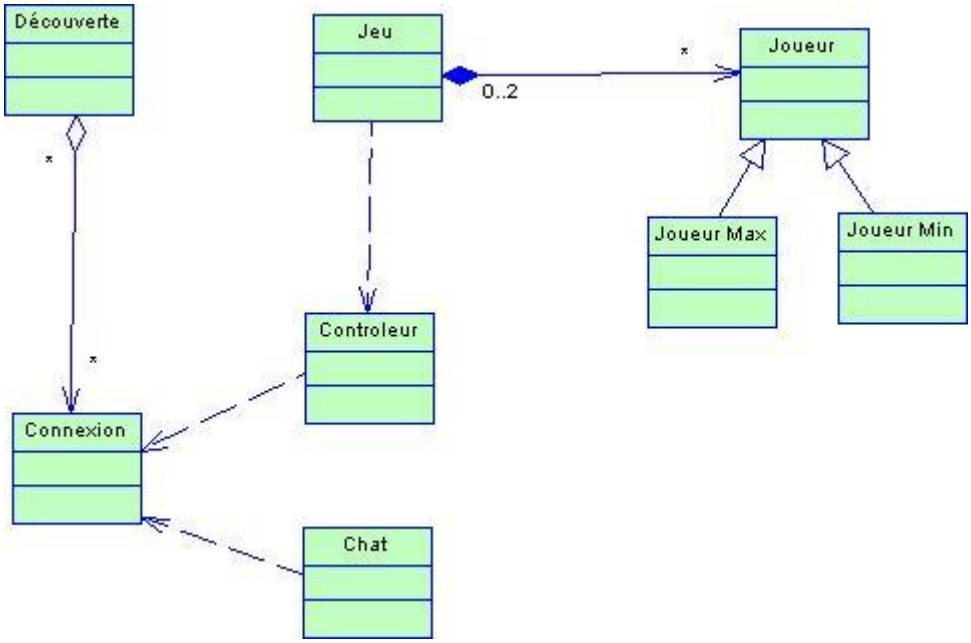


Figure 11 : Diagramme de Classes

6. Découpage en catégories

Cette phase d'analyse utilise la notion de package pour définir des catégories de classes d'analyse et découper le modèle UML en blocs logiques les plus indépendants possibles.

Une catégorie consiste en un regroupement logique de classes à forte cohérence interne et faible couplage externe.

Le découpage en catégories de notre projet a donné le résultat suivant :

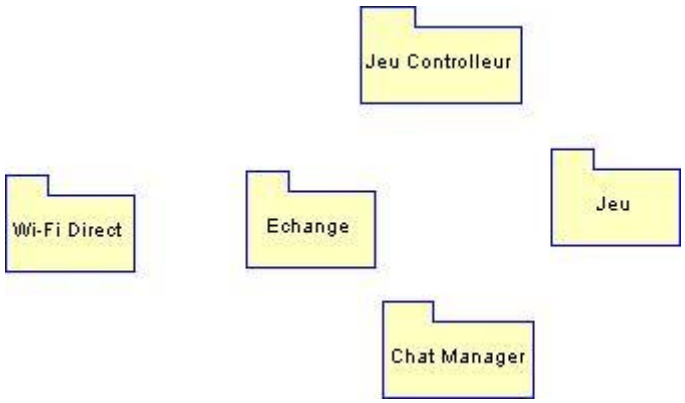


Figure 12 : Diagramme de packages

VI. Conclusion

Dans ce chapitre, nous avons présenté la modélisation de notre système. Nous avons établi un modèle de base caractérisant notre application. Dans ce chapitre, nous avons aussi, effectué une étude détaillée pour la mise en œuvre de notre système. La mise en place de ce dernier sera présentée dans le chapitre suivant.

.

Chapitre 3 : Mode de communication Wi-Fi Direct

I. Introduction

Le Wifi Direct, ou bien le Wifi Peer to Peer, est une technologie permettant aux périphériques Android récents (version 4.1 ou plus récente) le partage de données via leur connexion Wi-Fi sans point d'accès intermédiaire (routeur).

Ce protocole de transfert de données permet de découvrir et se connecter à d'autres périphériques comme on le ferait en Bluetooth mais à un débit bien plus élevé et sur des distances beaucoup plus longues.

Le Wi-Fi de l'appareil est facilement utilisable à partir d'une application qui utilise le package Wi-Fi P2P ainsi que tout ses méthodes.

Au cours de ce chapitre nous allons expliquer les différentes étapes de l'implémentation du WI-FI direct dans une application Android de façon bien détaillé et nous allons aussi nous intéresser au plus près à la programmation réseaux (sockets, threads….).

II. Les méthodes fournis par l'API

L'API « WifiP2pManager » de Google contient plusieurs méthodes :[4]

- initialize () : c'est la première méthode à appeler, elle enregistre l'application dans le cadre d'une connexion WI-FI.
- connect() : démarre la connexion Peer-to-Peer avec appareil dont la configuration est spécifié.
- requestConnectInfo() : Demande dispositif d'informations de connexion.
- Create group() : Crée un groupe de peer-to-peer avec le dispositif actuel en tant que propriétaire du groupe.
- removeGroup() : Supprime le groupe a0ctuel Peer-to-Peer.
- discoverPeers() : lance la recherche des périphérique.

III. La création d'une connexion Wi-FiP2P [4].

Création d'une application P2P Wi-Fi consiste à créer et enregistrer un récepteur de radiodiffusion pour l'application, à la découverte des périphérique, la connexion, ainsi que le transfert de données.les description suivante décrivent comment établir cela.

1. La configuration initiale [4]

Demander la permission d'utiliser le matériel Wi-Fi sur l'appareil et également déclarer votre demande d'avoir la version SDK minimum correct dans le manifeste Android

2. Vérification de l'état du Wi-Fi [4]

La classe implémenté WiFiDirectBroadcastReceiver nous informe des événements importants du wifiP2P

3. Détecter les appareils connectés [4]

Pour détecter les autres périphériques disponibles on utilise la méthode discoverPeers ().

Le WIFI_P2P_PEERS_CHANGED_ACTION sera informé pour mettre à jours la liste des « périphériques détecté »

4. Connexion a un périphérique [4]

Une fois le périphérique désiré trouver on fait appelle à la méthode connect() (méthode de connexion a l'appareil), qui nécessite un appel WifiP2pConfig comme objet a fin de récupéré les informations de l'appareil a connecter.

5. Transfert des données [4]

Une fois connexion établie, le transfert de donnée devient possible et cela en utilisant la programmation réseaux (multithread) en utilisant principalement les sockets qui sont eux même gérer par les threads

a. Rôles des sockets :

➔ Connexions à une machine distante : comme la connexion de deux ordinateurs sur un réseau

➔ Envois/Réception de données : comme le FTP et les messages

➔Fermeture d'une connexion

➔Attachement à un port

➔Acceptation d'une demande de connexion a un port local

➔Attente de demandes de connexion

b. Les différente Méthodes des sockets émission/réception de données :

- Accept () : attente de connexion d'un client distant, quand connexion est faite retourne une socket Permettant de communiquer avec le client : socket de service
- Close() : Ferme la socket et libère le port a laquelle elle était liée
- GetLocalPort() : Retourne le port local sur lequel est liée la socket
- getInputStream () : Retourne un flux d'entrée permettant de recevoir des donnée via la socket.
- getLocalAddress () : Retourne l'adresse IP locale de cette socket
- getLocalPort () : Retourne le port au quel cette socket est relié
- getLocalSocketAddress () : Retourne l'adresse locale et le port de cette prise en SocketAddress ou null si la prise est reliée
- getOutputStream () : Retourne un flux de sortie pour écrire des données dans cette prise
- BufferedReader : cette classe permet de lire des caractères à partir d'un flux tamponné, afin de faire des lectures plus rapides
- nputStreamReader : convertit un flux binaire en flux de caractères : elle convertit un objet de type InputStream en objet de type Reader
- PrintWriter ajoute à un flux la possibilité de faire des écritures sous forme de texte des types primitifs Java, et des chaînes de caractères
- getReceiveBufferSize () : retourne cette socket

c. Exemple simple de communication entre les sockets (Server/Client)

Le schéma ci-dessous montre comment s'effectue l'échange entre les sockets :

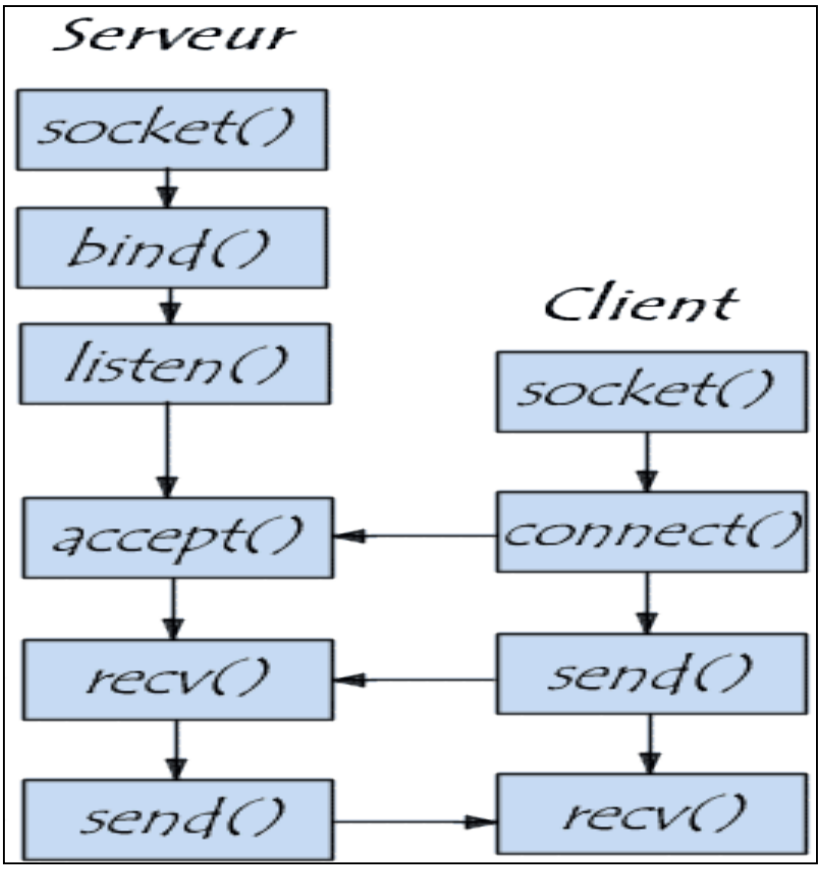


Figure 13:Exemple de communication entre les sockets

Le code pour implémenté cette communication

```java
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;

public class Serveur {

    public static void main(String[] zero) {

        ServerSocket socketserver  ;
        Socket socketduserveur ;

        try {

            socketserver = new ServerSocket(2009);
            socketduserveur = socketserver.accept();
            System.out.println("Un zéro s'est connecté !");
                socketserver.close();
                socketduserveur.close();

        }catch (IOException e) {
            e.printStackTrace();
        }
    }

}
```

Figure 14:Server socket [8]

```java
import java.io.IOException;
import java.net.InetAddress;
import java.net.Socket;
import java.net.UnknownHostException;


public class Client {

    public static void main(String[] zero) {

        Socket socket;

        try {

             socket = new Socket(InetAddress.getLocalHost(),2009);
                 socket.close();

        }catch (UnknownHostException e) {

            e.printStackTrace();
        }catch (IOException e) {

            e.printStackTrace();
        }
    }

}
```

Figure 15:client socket [8]

d. Transfère de donné en utilisant les sockets et les threads[4]

- ✓ Création d'un ServerSocket qui sera envoyé comme code d'exécution d'une thread. Cette prise attend une connexion d'un client sur un port et bloque jusqu'à qu'un client se connecte a ce port en utilisant la méthode accepte ().
- ✓ Création d'un client Socket qui sera de même envoyé comme code d'exécution d'une autre thread. ce client socket utilise l'adresse IP et le port du socket serveur pour se connecter à l'appareil de serveur.
- ✓ Le socket serveur attend une connexion client (avec la méthode accept()). Cet appel blocs jusqu'à ce qu'un client se connecte, lorsqu'une connexion se produit, le dispositif serveur peut recevoir les données à partir du client.
- ✓ Le code implémenté ci-dessous montre comment effectué un transfères a partir d'un client a un server :

Lorsqu'on clique sur le bouton du client, la chaîne de caractères à traiter (renverser) doit être envoyée au serveur

dans le code de la méthode onClick(), on écrit du code réseau d'envoi = une thread à créer et à lancer :

Sachant que les mise a jours graphique seront faite par un Handler (une thread relié a la User Interface thread)

Le thread ouvre un socket vers le serveur, écrit dedans, puis écoute cette même socket pour recevoir le résultat traité par le serveur :

IV. Conclusion

Dans ce chapitre nous avons expliqué comment implémenté une application qui permet d'interagir avec le WI-FI direct du périphérique fonctionnant sous Android a fin de pouvoir effectué les différente tache désiré tel que le transferts de données

Chapitre 4 : Mise en place et Implémentation

I. Introduction

Nous avons vu dans les chapitres précédents en quoi consiste notre application, et dans cette dernière partie nous allons nous intéresser aux outils utilisés pour sa réalisation et montrer ses principales interfaces.

II. Environnement du travail

1. Environnement matériel

- Deux PC portable TOSHIBA et HP
- Deux Smartphones Android version 4.1

2. Atelier de génie logiciel

a. Langage de programmation

Java : est le langage de programmation que nous avons utilisé pour notre application, et ceci pour ses multiples avantages, principalement pour sa portabilité il tourne sur une machine virtuelle par conséquent il peut fonctionner sur pc, Apple et même des appareils mobiles, sinon aussi pour sa performance et sa vitesse d'exécution, C'est le langage le plus adopté par les développeurs grâce à sa fiabilité et sa performance élevé.

b. Environnement de développement

- **JDK 1.7 (Java Développent Toolkit) :** Il contient d'une part le JRE (Java Run-time Environnements) c'est à dire les composants nécessaires au lancement d'applications Java ainsi qu'une machine virtuelle Java, et d'autre part un ensemble d'outils de programmation (javac, javadoc, etc.).

c. Conception

- **PowerDesigner :** C'est un logiciel de modélisation. Il permet de modéliser les traitements informatiques et leurs bases de données associées. il dispose d'un environnement graphique pour la saisie et la simulation de fonctionnement.

d. IDE : (Environnement de développement intégré)

- **ECLIPSE :** Eclipse est un environnement de développement modulable développé par IBM, capable de créer des programmes dans de nombreux langages de programmation (Java, C++, PHP…).

III. Scénarios Applicatif

Dans cette partie on va présenter notre version concrète du projet qui a été scindé en deux applications distinctes.

Une de type monoposte qui est présenté sous forme d'un jeu de Nim se jouant entre deux joueur à tour de rôle le joueur ayant retiré la dernière allumette sera le gagnant avec la possibilité de généré l'arbre du jeu qui permet d'indiquer les coups gagnant à l'aide de l'algorithme MiniMax. Et la deuxième est l'application 2AS Zalamites, qui est une application mobile.

V. La version monoposte :

Ci-dessous la première fenêtre de notre application qui offre la possibilité à l'utilisateur de lancer une nouvelle partie, voir les règles du jeu ou bien quitter.

Figure 16: Fenêtre principale

Cette fenêtre d'initialisation s'affiche après le clic du premier bouton « Nouvelle partie »

Figure 17: Fenêtre des paramètres du jeu

Après avoir définie les paramètres du jeu l'interface du déroulement de la partie apparait.

Figure 18:Interface déroulement du jeu

En cliquant sur le bouton « Help » l'arbre généré par l'algorithme MiniMax apparait (figure 17) pour nous montrer quel cout est gagnant, le nœud perdant est marqué « -1 » et le nœud gagnant est « 1 »

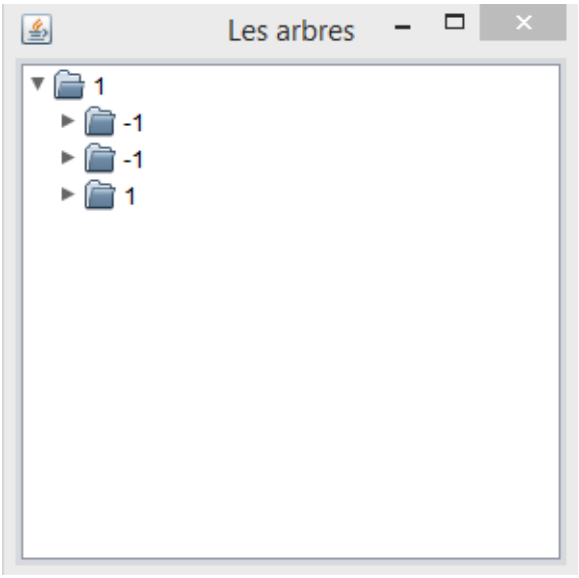


Figure 19 L'arbre MiniMax

VI. La version mobile :

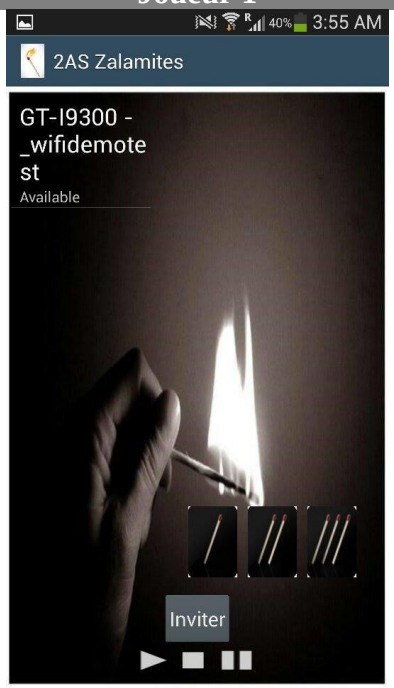
Ci-dessous les différentes interfaces de cette version mobile par ordre :

Recherche d'adversaire de jeu.

Joueur 1 Joueur 2

Liste de joueur trouvé après la recherche établie

Joueur 1 **Joueur 2**

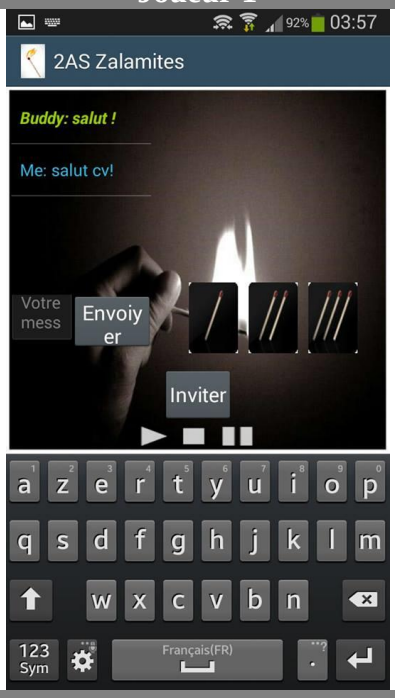


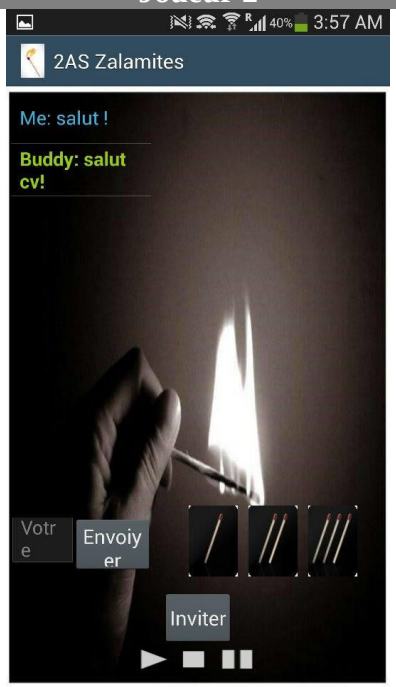
Réception d'invitation de connexion

Joueur 1 **Joueur 2**

Dans le cas d'acceptation de l'invitation, la 2eme interface se charge

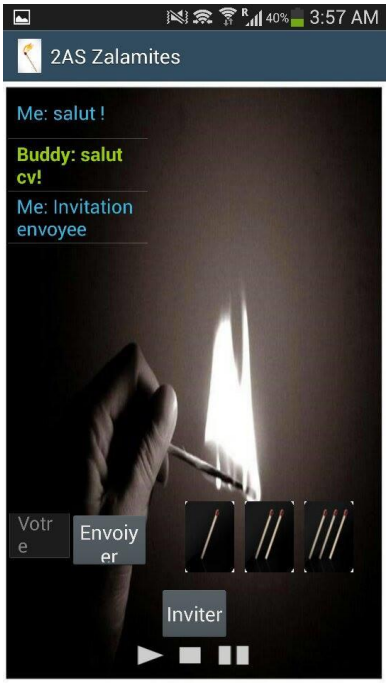
Joueur 1 **Joueur 2**

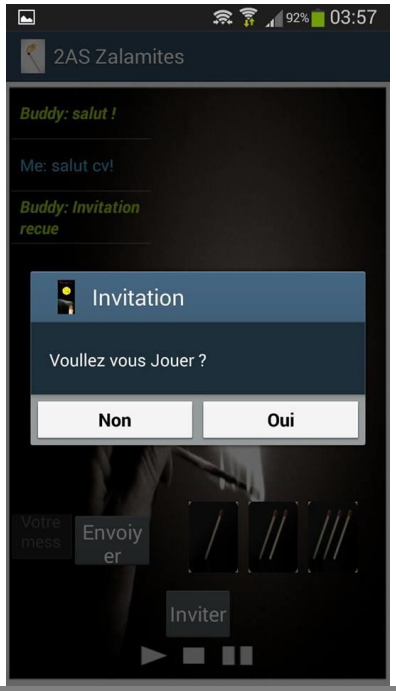




Réception d'invitation au jeu

Joueur 1 **Joueur 2**



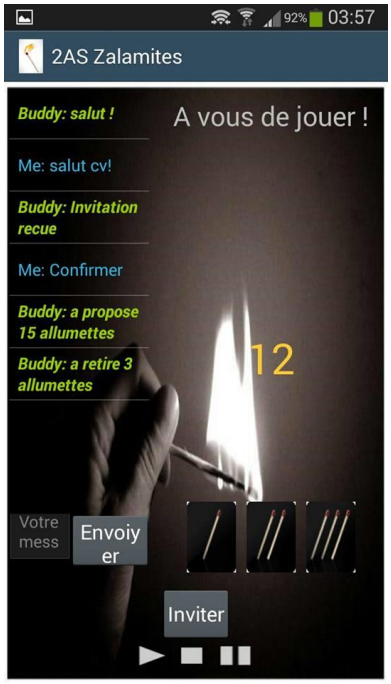


Initialisation de nombre d'allumettes

Joueur 1 **Joueur 2**

Déroulement d'une partie de jeux et l'historique du chat

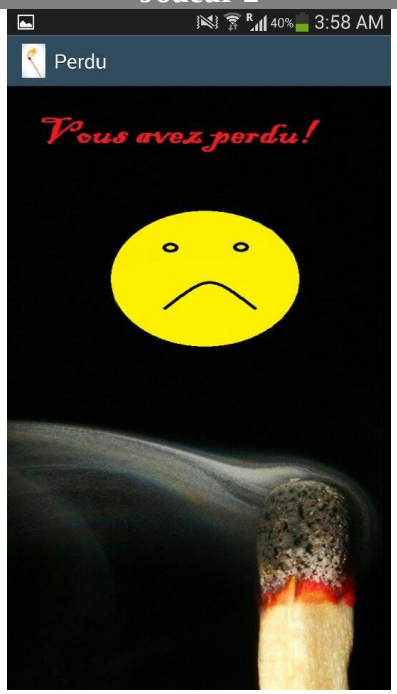
Joueur 1 **Joueur 2**



Fin de partie

Joueur 1

Joueur 2



Conclusion Générale

Nous avons donc réalisé une application Android contenant les fonctionnalités désirés. Ce projet nous a permis de découvrir le SDK Android, avec ses particularités. Cette expérience fut très enrichissante

Car la programmation sous Android permet d'utiliser nos connaissances en JAVA dans un nouveau contexte qui est aujourd'hui présent sur de nombreuses plateformes portables (tablettes, Smartphone...etc.). Ainsi notre application est potentiellement portable sur plusieurs type d'appareils fonctionnant sous Android, ce qui permettrait un maillage plus grand.

Ce projet nous a permis de mettre en place l'algorithme de parcours intelligent d'arbre de jeu, en l'occurrence l'algorithme MiniMax.

L'aspect concret de ce projet nous a plu car nous pouvions voir facilement le résultat de notre travail, ce qui fut très motivant.

L'application réalisée peut subir des améliorations plus ou moins intéressantes ; dans ce contexte nous envisageons d'implémenter le module de communication audio entre joueurs, aussi et comme deuxième perspective, nous envisageons étendre l'application pour qu'elle puisse supportée un jeu à multi-joueurs (plus de deux joueurs).

Bibliographie

1 Jean Kruger, « Opportunités d'entreprise avec Android », Institut SupInfo, 2009. Mémoire de fin d'études

2 Khaoula Mrabet, Nessrine Trabelsi, « Application de messagerie simple sur Android », Télécom SudParis, 2011.

3 http://turing.cs.pub.ro/auf2/html/chapters/chapter3/chapter_3_4_2.html

4 http://developer.android.com/guide/topics/connectivity/wifip2p.html

5 http://fr.wikipedia.org/

6 Pascal Roques Franck Valeéé « UML 2 en action »,4e édition

7 http://uml.free.fr/

8 http://fr.openclassrooms.com/informatique/cours/introduction-aux-sockets-1